Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

I-ACCESS

Implementing the Accessibility to Urban Historic Center's Use and Knowledge

F. Giuliano, G. Rizzo, A. Scianna,

Rapporto Tecnico N.: 2
RT-ICAR-PA-20-01 **novembre 2020**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Cosenza, Via P. Bucci Cubo 8/9C, 87036 Rende, Italy, URL: *www.icar.cnr.it*
– Sede di Napoli, Via P. Castellino 111, 80131 Napoli, URL: *www.na.icar.cnr.it*
– Sede di Palermo, Via Ugo La Malfa 153, 90146 Palermo, URL: *www.pa.icar.cnr.it*

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

I-ACCESS

Implementing the Accessibility to Urban Historic Center's Use and Knowledge

F. Giuliano, G. Rizzo, A. Scianna,

Rapporto Tecnico N.:2 ***Data:***
RT-ICAR-PA-20-01 **novembre 2020**

Project: “I-ACCESS Implementing the Accessibility to Urban Historic Center’s Use and Knowledge”

INDICE DEI CONTENUTI

1. *Introduzione*

La seguente relazione descrive le attività condotte ed i risultati ottenuti tra giorno 03/06/2019 ed il giorno 31/10/2019. Il lavoro svolto è stato principalmente orientato alla progettazione e realizzazione di un'architettura software dedicata all'organizzazione di informazioni geo-referenziate di accessibilità del patrimonio culturale oggetto del progetto I-ACCESS. Si è proceduto alla progettazione di un sistema informativo dedicato allo scopo individuando delle caratteristiche di flessibilità e modularità tali da renderlo facilmente replicabile ed estensibile. Il sistema è stato implementato sotto forma di servizio Web, sfruttando esclusivamente tecnologie a sorgente aperto ed esponendo delle interfacce Web standard tali da essere integrabili con applicazioni di terze parti per mezzo di opportune Application Programming Interfaces (API). Una volta definito ed implementato il framework principale del servizio web, è stata implementata un'applicazione definita "Piattaforma Collaborativa", la quale ha come obiettivo quello di fornire uno strumento per gestire l'informazione di accessibilità dei luoghi di interesse attraverso delle notifiche inoltrate dagli utenti. Infine è stata progettata l'interfaccia di comunicazione di un sistema di lettura in real-time dei punti annunciate attraverso tecnologie radio a bassa potenza basata su Bluetooth (Bluetooth Low Energy – BLE).

2. *Descrizione architettura Web-Service*

In questo paragrafo è descritta l'architettura del sistema composta da i) le funzioni logiche principali del servizio che si intende offrire, ii) una descrizione tecnica dettagliata delle tecnologie proposte.

L'intero sistema è stato sviluppato usando soluzioni open-source tipiche delle applicazioni orientate al Web. Al fine di fornire la messa in campo di un sistema flessibile, è stato deciso di sfruttare architetture di virtualizzazione di sistemi di varia natura (Web-server, microservizi, Database,…), racchiusi in componenti chiamati "container". Tali "container" possono operare in modo indipendente e coesistere in modo trasparente alla configurazione specifica del Sistema Operativo, questo consente di rendere replicabile l'installazione dell'intero servizio Web su macchine diverse indipendentemente dalla natura del sistema operativo stesso. Nel progetto I-ACCESS è stato proposto di implementare l'intero sistema sfruttando il framework chiamato Docker[1]; esso è infatti uno dei più famosi progetti che permette l'implementazione di prodotti di tipo Platform-as-a-Service (PaaS) e Software-as-a-service (SaaS). Docker offre una serie di utili repository dove potere scaricare delle cosiddette "immagini" di servizi base (e.s. MySQL[2], PostgreSQL[3], MongoDB[5], PHP[6], Apache[7], NGNIX[8], python-Flask[9], python-Django[10] e molti altri) ed estenderli in relazione alle esigenze specifiche del prodotto che si intende realizzare. La composizione di più servizi è fatta attraverso degli strumenti offerti dal framework Docker, uno fra tutti è

docker-compose, uno strumento che consente di installare in modo compatto e diretto tutti i servizi desiderati.

Nel progetto I-ACCESS il servizio web che si intende realizzare individua due elementi principali: un Database Management system (DBMS) e un insieme di "Microservizi" che espongono le API, tipicamente di tipo Http REST.

Database Management systems (DBMS)

In merito ai sistemi per la gestione dei Database (DBMS), l'attuale stato dell'arte individua due princiapli famiglie di sistemi: Database **Relazionali (SQL)** e database **Non (solo) Relazionali (NoSQL).** I Database relazionali (e.s. MySQL o PostgresSQL), sono quei sistema in cui i dati sono logicamente organizzate in tabelle, dove ogni riga contiene un numero di colonne prestabilito. L'accesso ai dati è basato su in linguaggio definito Structured Query Language (SQL). Questi tipi di Database forniscono strumenti solidi per la gestione e l'interazione di database composti da molte tabelle, le quali sono tipicamente in relazione tra loro. Una tabella può essere collegata alle altre attraverso specifichi record definiti "Chiavi". Tale approccio permette di progettare un schema di dati coerente e ben strutturato.

I principali vantaggi dei Database Relazionali sono:

- *Modello dati Semplice*: Orientato alla tabella, facile da implementare e gestire, utile nel caso in cui i dati devono essere memorizzati per un lungo periodo di tempo;
- *Bassa Ridondanza*: I database Relazionali mirano a ridurre le ridondanze, importante vantaggio specialmente nella fase di gestione dei dati, grazie all'utilizzo di tecniche come l'isolamento dei dati.
- *Elevata coerenza dei dati*: I Database Relazionali definiscono delle funzionalità automatiche che escludono quelle transazioni che possono causare problemi di coerenza dei dati (e.s chiavi duplicate);
- *Orientate all'elaborazione di grandi quantità di dati*: Nei database relazionali ogni entità (tabella) è composta da elementi atomici (colonne), questo consente di definire specifiche correlazioni tra entità diverse consentendo di eseguire in modo agevole chiamate complesse come la JOIN.
- *Linguaggi unificato e standardizzato*: SQL è un linguaggio standardizzato e, in linea principio, premette di avere un'implementazione agonistica ed essere compatibile con diverse versioni di DBMS (e.s MySQL, PostGres, etc…). Tuttavia, in applicazioni pratiche diversi DBMS possono avere alcune differenze nella definizione del linguaggio che non permette la completa compatibilità delle query.

In molte applicazioni i DBMS lavorano in stretta sinergia con linguaggi di programmazione orientata agli oggetti e molto spesso le strutture dati sono essenzialmente definite attraverso un oggetto. In questi casi il DBMS dovrebbe essere in grado di convertire l'oggetto in una Relazione sul database, questa operazione prende il nome di Object Relation Mapping (ORM). Tuttavia, in alcune applicazioni, la definizione di uno schema fisso del database relazionale può causare dei cosiddetti "mismatch" tra oggetti ed istanze del db, causando errori di consistenza, Al fine di risolvere questo inconveniente, sono stati introdotti i cosiddetti Not-Only-Relational Database, vale a dire i Database Non Relazionali. Il principale vantaggio consiste nel fatto che l'assenza di una struttura predefinita della tabella consente di non incorrere a problemi di correlazione tra un oggetto ed un record del database. Uno dei più utili e famosi Database Non relazionali è MongoDB.

Durante l'analisi delle specifiche di progettazione del sistema, uno dei requisiti fondamentali è legato al supporto dei cosiddetti "Geographical Information System" (GIS), uno dei più DBMS open-source più conosciuti è PostgresSQL, il quale è dotato di una specifica estensione chiamata PostGIS. con PostGIS è possibile effettuare delle cosiddetta "*Spatial Queries*" vale a dire delle interrogazioni che fanno riferimento alle caratteristiche geografiche del dato. Nonostante sia DBMS SQL che NoSQL possano supportare servizi GIS la scelta progettuale concordata tra i partner è stata di utilizzare PostGIS, che sembra essere la più stabile e collaudata.

Microservizi – definizioni delle API

L'approccio di progettazione del sistema informativo di I-ACCESS proposto è basa su microservizi. Un'architettura di microservizi consente di disaccoppiare le funzionalità dell'intero servizio in attività modulari, flessibili e con scopi specifici. L'architettura proposta definisce un microservizio principale in cui è implementata l'interfaccia di comunicazione tra Database e applicazioni esterne (API) basata su chiamate HTTP Representational State Transfer (REST). Per quanto riguarda l'implementazione dell'API, la flessibilità dell'intero sistema è in grado di definire diversi microservizi utilizzando framework e linguaggio di programmazione diversi. In questa fase, le attività del progetto sono orientate su un'implementazione su un framework di microservizi sviluppato in python chiamato Flask; questo framework è molto compatto e può sfruttare tutte le librerie python offerte per l'analisi dei dati e consente di distribuire un'applicazione con un utilizzo minimo del codice, snella ed efficiente.

Una descrizione dettagliata degli elementi dell'architettura I-ACCESS è mostrata in Figura 1.

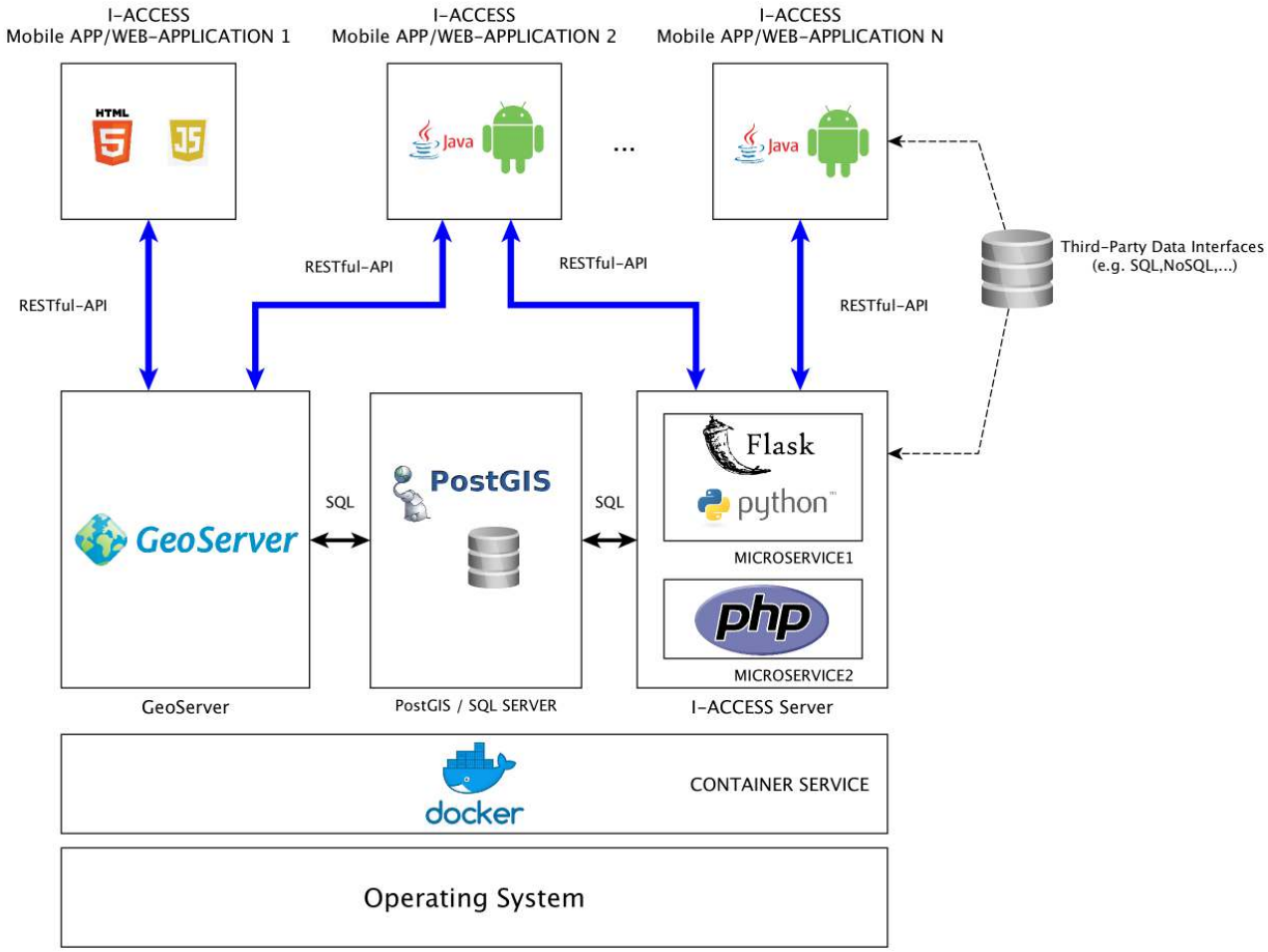

Figura 1: Architettura della piattaforma I-ACCESS

L'ambiente del sistema operativo prevede l'installazione di un servizio Docker Container il quale ha il compito di orchestrare tutti i servizi ospitati sul server. Nell'architettura proposta, possiamo distinguere tre *container* principali: il primo *container* è GeoServer, una soluzione matura, solida e affidabile per integrare una complessa struttura di dati GIS. Consente la definizione di più livelli (ad es. Punti di interesse, percorsi, edifici, ecc.), e l'integrazione con sorgenti dati eterogenee. GeoServer è una suite completa che include e include diverse opzioni di presentazione dell'utente, utili per velocizzare la presentazione e la visualizzazione dei livelli. Un'altra potente funzionalità fornita da GeoServer è la definizione di sofisticate query spaziali che consentono di adattare l'operazione di ricerca nell'area geo-referenziata, migliorando le prestazioni e fornendo informazioni importanti correlate al contesto geografico. Tutte le funzionalità di GeoServer sono gestite ed esposte da un ampio set di API documentate, offrendo uno strumento pratico per eseguire richieste RESTful da parte delle applicazioni. Un altro *container* è identificato dal Database che include un server PostGIS. Il server PostGIS ospiterà tutti i database, le tabelle e le relazioni tra le

informazioni geografiche, che saranno progettati al fine integrare le informazioni geografiche sul contesto dell'accessibilità, includerà un insieme di dati sulle informazioni generali come orari di apertura o logistica informazioni di contatto.

Oltre a GeoServer e PostGIS il terzo *container* dell'architettura proposta è il server "I-ACCESS". Questo server è costituito da un diversi *container* relativi a diversi microservizi. Un microservizio è progettato per rispondere a requisiti specifici e offre un sottoinsieme personalizzato di funzionalità, che può essere esposto a una o più applicazioni. Questo approccio consente di accelerare lo sviluppo, riduce le fasi di test e debug e riduce le interruzioni di servizio. Ogni microservizio espone le sue funzionalità definendo un'API in forma compatta e ben focalizzata su uno specifico campo di attività. Il progetto prevede la definizioni dei due microservizi: Il primo microservizio, chiamato "MICROSERVICE1" è espone una serie di API REST utili per estrarre e manipolare informazioni GIS, accessibilità dei punti di interesse; il secondo microservizio è "MICROSERVICE2" ed espone un web server APACHE-PHP.

3. Definizione Modello Dati del sistema

Il modello dati è di tipo Entità-Relazione (ER) e sintetizza i requisiti fondamentali dei punti di interesse i quali possono essere riassunti in: a) Organizzare e sintetizzare le informazioni sull'accessibilità di un luogo, b) Organizzare e sintetizzare le informazioni sul Punto di interesse, c) collegare descrizioni e supporti aggiuntivi a ciascun Punto di interesse, d) Organizzare le informazioni sul Percorso e il rapporto tra percorsi e Punto di Interesse.

In base all'analisi dei requisiti, possiamo definire le seguenti entità principali: **POI (Point of Interest)**, che organizzano le luoghi di interesse culturale in modo geo-referenziato evidenziando i principali campi informativi; **Accessibilità (Accessibility)**, che estende le informazioni sui **POI** introducendo si informazioni generali (ad es. Telefono, indirizzo, URL del sito Web, orari di apertura, ...), si informazioni specifiche per i disabili al fine di fornire informazioni rapide e sintetiche sull'accessibilità dei luoghi; **Itinerari (ROUTES)**, che contiene percorsi geo-referenziati che possono collegare diversi **POI**;, infine, **POI_media** consente di arricchire il descrizione con contenuti multimediali, come immagini, audio, video o qualsiasi altra risorsa di realtà aumentata (AR). Le sezioni seguenti descrivono in dettaglio ogni entità e ciascun campo.

Il modello dati proposto è indicato nella Figura 2

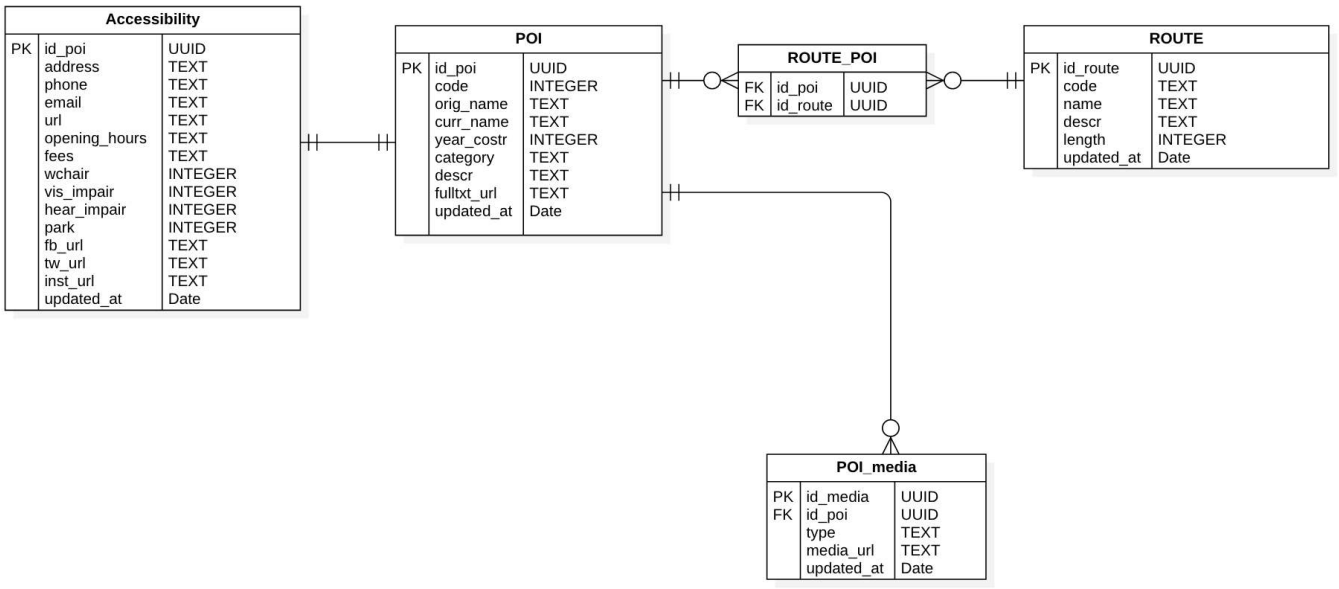

Figura 2 - Modello Dati Entità Relazione

4. Descrizione delle Entità

4.1. POI

Point of Interest Identification [id_poi]

Description	Quetso campo è univocamente generato seguendo lo standard **Universally unique identifier** (**UUID**), fornito da Open Software Foundation (OSF).
Type:	UUID
Field values	Campo richiesto. Esso viene solitamente generato dal software che colleziona i dati. QGIS software, per esempio, offre a una funzione chiamata uuid() che genera una stringa come la seguente: `550e8400-e29b-41d4-a716-446655440000`

Arbitrary Code Indicator [code]

Description	Codice arbitrario che può essere assegnato dal gestore del sistema per usi futuri
Type:	INTEGER
Field notes	Opzionale, può essere riempito con NULL

Point of Interest Origin Name [orig_name]

Description	Questo campo include il nome originario del punto di interesse
Type:	TEXT
Field notes	Opzionale, può essere riempito con NULL

Point of Interest Current Name [orig_name]

Description	Questo campo contiene il nome del punto di interesse
Type:	TEXT
Field notes	Campo richiesto

Short Description [Descr]

Description	Breve descrizione del Punto di interesse, può includere anche tag ipertestuali, utili per favorire tecniche di indicizzazione o analisi semantica
Type:	TEXT
Field notes	Opzionale, può essere riempito con NULL

Year of Construction [year_costr]

Description	Contiene la data o l'anno di realizzazione dell'edificio o del monumento associato al punto di interesse.
Type:	Date
Field notes	Opzionale ma altamente consigliato. Se la data precisa non è nota, ma è noto solo l'anno, è possibile fornire indicazioni parziali (e.s. 1-1-1570)

Category

Description	Questo campo deve contenere la categoria del punto di interesse
Type:	TEXT
Field notes	Opzionale ma altamente consigliato. Il campo può essere validato a NULL, si suggerisce di inserire una o due parole.

Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio Nazionale delle Ricerche (ICAR-CNR)Via P. Bucci cubo 8/9C, 87036 Rende (CS) – Via P. Castellino 111, 80131 Napoli – Via U. La Malfa 153, 90146 Palermo Partita IVA 02118311006 – Codice Fiscale 80054330586 – protocollo.icar@pec.cnr.it – www.icar.cnr.it

Point of interest last update [updated_at]

Description	Indica la data di aggiornamento delle informazioni relative al POI, utile per tenere traccia di eventuali aggiornamenti del punto di interesse.
Type:	Date
Field notes	Campo autogenerato. Ad ogni aggiornamento il campo viene aggiornato con nuova data ed ora.

4.2. Accessibility

Point of Interest Identification [id_poi]

Description	Quetso campo è univocamente generato seguendo lo standard **Universally unique identifier** (**UUID**), fornito da Open Software Foundation (OSF).
Type:	UUID
Field values	Campo richiesto. Esso viene solitamente generato dal software che colleziona i dati. QGIS software, per esempio, offre a una funzione chiamata uuid() che genera una stringa come la seguente: `550e8400-e29b-41d4-a716-446655440000`

Address [address]

Description	Contiene la descrizione dell'indirizzo il numero civico e il CAP.
Type:	TEXT
Field values	Campo richiesto

Phone [phone]

Description	Numero Telefonico del gestore del POI
Type:	TEXT
Field values	Opzionale ma altamente consigliato. Il campo può essere validato a NULL.

E-mail address [email]

Description	E-mail address del gestore del POI
Type:	TEXT
Field values	Opzionale ma altamente consigliato. Il campo può essere validato a NULL.

Point of Interest website URL [url]

Description	Indirizzo web del POI
Type:	TEXT
Field values	Opzionale ma altamente consigliato. Il campo può essere validato a NULL.

Point of interest opening hours [opening_hours]

Description	Orari di apertura del POI
Type:	TEXT
Field values	Opzionale ma altamente consigliato. Il campo può essere validato a NULL.

Point of Interest Fees [fees]

Description	Questo campo contiene il costo per visitare al POI
Type:	TEXT
Field values	Opzionale ma altamente consigliato. Il campo può essere validato a NULL.

Accessibility indication for physically impaired persons [wcharis]

Description	Questo campo contiene il livello di accessibilità del POI per persone con disabilità fisica.
Type:	INTEGER
Fields values	Opzionale ma altamente consigliato. Il campo può essere validato a NULL. Questo campo può avere la seguente convenzione:

0 – RED	no accessibility

	1 – ORANGE:	partial accessibility (for example in some area of the Point of Interest)
	2- GREEN:	full accessibility.

Accessibility indication for visually impaired persons [vis_impair]

Description	Questo campo contiene il livello di accessibilità del POI per persone con disabilità visiva.
Type:	INTEGER
Fields values	Opzionale ma altamente consigliato. Il campo può essere validato a NULL. Questo campo può avere la seguente convenzione:

0 – RED	no accessibility
1 – ORANGE:	partial accessibility (for example in some area of the Point of Interest)
2- GREEN:	full accessibility.

Accessibility indication for hearing impaired persons [hear_impair]

Description	Questo campo contiene il livello di accessibilità del POI per persone con disabilità uditiva.
Type:	INTEGER
Fields values	Opzionale ma altamente consigliato. Il campo può essere validato a NULL. Questo campo può avere la seguente convenzione:

0 – RED	no accessibility
1 – ORANGE:	partial accessibility (for example in some area of the Point of Interest)
2- GREEN:	full accessibility.

Car Parking accessibility [park]

Description	Questo campo contiene il livello di accessiblità di parcheggio per gli automezzi.

Type:	INTEGER
Fields values	Opzionale ma altamente consigliato. Il campo può essere validato a NULL. This field It is defined like a semaphore: 0 – RED: Nessuna accessibilità 1 – BLUE: Riservato solo per invalidi 2- GREEN: Parcheggio disponibile per tutti

fb_url

Description	URL Facebook del gestore o dell'organizzazione associata al POI.
Type:	TEXT
Field values	Opzionale ma altamente consigliato. Il campo può essere validato a NULL.

tw_url

Description	URL Twitter del gestore o dell'organizzazione associata al POI.
Type:	TEXT
Field values	Opzionale ma altamente consigliato. Il campo può essere validato a NULL.

inst_url

Description	URL Instagram del gestore o dell'organizzazione associata al POI.
Type:	TEXT
Field values	Opzionale ma altamente consigliato. Il campo può essere validato a NULL.

Point of interest last update [updated_at]

Description	Indica la data di aggiornamento delle informazioni relative al POI, utile per tenere traccia di eventuali aggiornamenti del punto di interesse.
Type:	Date
Field notes	Campo autogenerato. Ad ogni aggiornamento il campo viene aggiornato con nuova data ed ora.

4.3. POI_media

Point of Interest media Identification [id_media]

Description	Quetso campo è univocamente generato seguendo lo standard **Universally unique identifier** (**UUID**), fornito da Open Software Foundation (OSF).
Type:	UUID
Field values	Campo richiesto. Esso viene solitamente generato dal software che colleziona i dati. QGIS software, per esempio, offre a una funzione chiamata uuid() che genera una stringa come la seguente: `550e8400-e29b-41d4-a716-446655440000`

Point of Interest Identification [id_poi]

Description	Quetso campo è univocamente generato seguendo lo standard **Universally unique identifier** (**UUID**), fornito da Open Software Foundation (OSF).
Type:	UUID
Field values	Campo richiesto. Esso viene solitamente generato dal software che colleziona i dati. QGIS software, per esempio, offre a una funzione chiamata uuid() che genera una stringa come la seguente: `550e8400-e29b-41d4-a716-446655440000`

Type

Description	Questo campo contiene il tipo del dato multimediale
Type:	TEXT
Field values	Opzionale ma altamente consigliato. Il campo può essere validato a NULL. Valori accettati: “AUDIO”,”IMAGE”, “VIDEO”, “AR”

media_url

Description	Questo campo contiene l’URL del dato multimediale.
Type:	TEXT
Field values	Opzionale ma altamente consigliato. Il campo può essere validato a NULL.

Point of interest last update [updated_at]

Description	Indica la data di aggiornamento delle informazioni relative al POI, utile per tenere traccia di eventuali aggiornamenti del punto di interesse.
Type:	Date
Field notes	Campo autogenerato. Ad ogni aggiornamento il campo viene aggiornato con nuova data ed ora.

4.4. ROUTES

Una funzionalità importante del progetto comprende la possibilità di fornire servizi di accessibilità basati su percorsi predefiniti e tour guidati. Per queste ragioni è stata definita un'entità chiamata **ROUTES**, la quale contiene informazioni geo-referenziate di specific percorsi che collegano fra lori i POI.

Route unique identifier [id_route]

Description	Quetso campo è univocamente generato seguendo lo standard **Universally unique identifier** (**UUID**), fornito da Open Software Foundation (OSF).
Type:	UUID
Field values	Campo richiesto. Esso viene solitamente generato dal software che colleziona i dati. QGIS software, per esempio, offre a una funzione chiamata uuid() che genera una stringa come la seguente: `550e8400-e29b-41d4-a716-446655440000`

Arbitrary Code Indicator [code]

Description	Codice arbitrario che può essere assegnato dal gestore del sistema per usi futuri
Type:	INTEGER
Field notes	Opzionale, può essere riempito con NULL

Route name [name]

Description	Nome assegnato al percorso, e.s. "Percorso Arabo Normanno".
Type:	TEXT
Field values	Campo richiesto

Route Description [descr]

Description	Descrizione sintetica del percorso che ne evidenzia che caratteristiche principali
Type:	TEXT
Field values	Campo richiesto

Route length [length]

Description	Contiene la lunghezza del percorso
Type:	Float
Field values	Non necessario, può essere auto-generato a partire dai dati geo-referenziati

ROUTE last update [updated_at]

Description	Indica la data di aggiornamento delle informazioni relative al POI, utile per tenere traccia di eventuali aggiornamenti del punto di interesse.
Type:	Date
Field notes	Campo auto-generato. Ad ogni aggiornamento il campo viene aggiornato con nuova data ed ora.

4.5. ROUTE_POI

Questa Entità permette di realizzre una relazione molti-a-molti POI e ROUTE, non ha nessuna funzione concettuale, ha solo funzione implementativa.

Route Identifier [id_route]

Description	Contiene l'identificativo di una ROUTE
Type:	UUID
Field values	Campo richiesto

Point of interest Identifier [id_poi]

Description	Contiene l'identificativo di un POI
Type:	UUID
Field values	Campo richiesto

4.6. Tabelle Addizionali

Il modello Dati descritto in Figura 2 può essere esteso con molte altre entità. In particolare un'entità di grande utilità è quella definita come ***"Monumental Buildings"*** . I campi distintivi della entità sono:

Unique identifier [gid]

Description	Quetso campo è univocamente generato seguendo lo standard **Universally unique identifier** (**UUID**), fornito da Open Software Foundation (OSF).
Type:	UUID
Field values	Campo richiesto. Esso viene solitamente generato dal software che colleziona i dati. QGIS software, per esempio, offre a una funzione chiamata uuid() che genera una stringa come la seguente: `550e8400-e29b-41d4-a716-446655440000`

Country code [ISO_3166_1]

Description	Standard country code basato sullo standard ISO 3166-1. Esempio: Per l'Italia il country code è 380, per Malta è 470
Type:	Integer
Field values	Maggiori informazioni: - https://www.iso.org/iso-3166-country-codes.html

Sub-area identification [district]

Description	Questo campo può essere differente per ogni territorio. A titolo di esempio, per la città di Palermo il comune definisce i distretti con il nome di "mandamento".
Type:	Integer
Field values	Opzionale

Initial building name [denom_ini]

Description	Al fine di avere informazioni estese riguardo l'edificio, questo campo contiene il nome originariamente assegnato all'edificio.
Type:	TEXT
Field values	Opzionale ma altamente consigliato. Il campo può essere validato a NULL.

Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio Nazionale delle Ricerche (ICAR-CNR)Via P. Bucci cubo 8/9C, 87036 Rende (CS) – Via P. Castellino 111, 80131 Napoli – Via U. La Malfa 153, 90146 Palermo Partita IVA 02118311006 – Codice Fiscale 80054330586 – protocollo.icar@pec.cnr.it – www.icar.cnr.it

Current building name [nome_att]

Description	Al fine di avere informazioni estese riguardo l'edificio, questo campo contiene il nome attuale dell'edificio.
Type:	TEXT
Field values	Opzionale ma altamente consigliato. Il campo può essere validato a NULL.

Short description [descriz]

Description	Una descrizione compatta che consente di fornire una descrizione sintetica della costruzione storica.
Type:	TEXT
Field values	Opzionale ma altamente consigliato. Il campo può essere validato a NULL.

Year building construction [anno_costr]

Description	Contiene la data o l'anno di costruzione dell'edificio storico.
Type:	Date
Field notes	Opzionale ma altamente consigliato. Il campo può essere validato a NULL.

Number of floors [n_piani]

Description	Contiene il numero di elevazioni dell'edificio.
Type:	INTEGER
Field notes	Opzionale. Il campo può essere validato a NULL.

Maximum height [h_max]

Description	Contiene l'altezza massima dell'edificio
Type:	INTEGER
Field notes	Opzionale. Il campo può essere validato a NULL.

Initial building usage [uso_ini]

Description	Contiene una breve stringa (preferibilmente una sola parola) relativa alla destinazione d'uso iniziale dell'edificio.

Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio Nazionale delle Ricerche (ICAR-CNR)Via P. Bucci cubo 8/9C, 87036 Rende (CS) – Via P. Castellino 111, 80131 Napoli – Via U. La Malfa 153, 90146 Palermo Partita IVA 02118311006 – Codice Fiscale 80054330586 – protocollo.icar@pec.cnr.it – www.icar.cnr.it

Type:	TEXT
Field values	Opzionale. Il campo può essere validato a NULL.

Current building usage [uso_att]

Description	Contiene una breve stringa (preferibilmente una sola parola) relativa alla destinazione d'uso attuale dell'edificio.
Type:	TEXT
Field values	Opzionale. Il campo può essere validato a NULL.

Owner [propriet]

Description	Nome del proprietario dell'edificio.
Type:	TEXT
Field values	Opzionale. Il campo può essere validato a NULL.

Operator [gestore]

Description	Nome del gestore dell'edificio
Type:	TEXT
Field values	Opzionale. Il campo può essere validato a NULL.

Building category [categoria]

Description	Questo campo deve contenere la categoria dell'edificio. Ad esempio: "Edificio pubblico" "Edificio privato", "Edificio religioso".
Type:	TEXT
Field notes	Opzionale. Il campo può essere validato a NULL.

Building type [tipo]

Description	Questo campo contiene una stringa compatta al fine di definire il tipo di edificio. Esempio: "chiesa", "caserma", "monastero".
Type:	TEXT

Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio Nazionale delle Ricerche (ICAR-CNR)Via P. Bucci cubo 8/9C, 87036 Rende (CS) – Via P. Castellino 111, 80131 Napoli – Via U. La Malfa 153, 90146 Palermo Partita IVA 02118311006 – Codice Fiscale 80054330586 – protocollo.icar@pec.cnr.it – www.icar.cnr.it

Field values	Opzionale. Il campo può essere validato a NULL.

Building structure type [tipstruel]

Description	Questo campo identifica la tipologia di struttura dell'edificio
Type:	TEXT
Field values	Opzionale. Il campo può essere validato a NULL.

Roofing type [copertura]

Description	Questo campo indica la tipologia di copertura dell'edificio.
Type:	TEXT
Field values	Opzionale. Il campo può essere validato a NULL.

5. *Implementazione Web-service*

L'implementazione del web-service è stata attuata in tre fasi: la prima fase è stata quella di implementare in modo distinto i servizi costituenti (Database, Microservizi e Geoserver) al fine di individuarne i requisiti principali. La seconda fase è stata di integrazione su container Docker. L'intero codice è disponibile su repository privato bitbucket. Il link per l'accesso è stato inviato, mediante invito alla condivisione del repository, al responsabile scientifico Andrea Scianna ed al collega Filippo Vella. Attualmente il link privato è disponibile a questo indirizzo: https://bitbucket.org/fabriziogiulianounipa/iaccess/

5.1. Installazione

Il progetto I-ACCESS è stato integrato su container Docker, per maggiori dettagli si invita ad analizzare il file docker-compse.yml presente all'interno del repository, in esso sono contenuti i dettagli implementativi. Per maggiori dettagli riguardo l'installazione di Docker si suggerisce di visitare le guide disponibili al seguente indirizzo: https://www.docker.com/

Il repository docker contiene uno script bash che riassume i passi da fare per installare Docker su sistemi Linux UBUNTU:

```
sudo apt-get update
sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent software-
properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo apt-key fingerprint 0EBFCD88
sudo add-apt-repository \
"deb [arch=amd64] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) \
test"
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io #docker-compose
#latest update: not install docker-compose for older ubuntu version (e.g. 16.04)
sudo curl -L "https://github.com/docker/compose/releases/download/1.24.0/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
#POST-INSTALLATION: Manage Docker as nn-root user
sudo groupadd docker
sudo usermod -aG docker $USER
```

Una volta scaricato il codice presente nel repository ed installata una versione recente di Docker, l'installazione del web-service di I-ACCESS avviene attraverso i seguenti passi:

```
git clone https://fabriziogiulianounipa@bitbucket.org/fabriziogiulianounipa/iaccess.git
> cd iaccess-webserver-docker
> docker-compose up --build –d #-d option to launch services in background
```

Se l'installazione viene completata con successo, i servizi verranno avviati sequenzialmente e dopo un transitorio di attivazione il sistema è pronto per lavorare.

Nota: Il sistema viene inizializzato senza installare nessun dato geo-referenziato, per tale ragione alcune interfacce restituiranno un'eccezione legata all'assenza delle tabelle. Il comportamento è del tutto normale, occorre solo inserire correttamente i dati attraverso l'endpoint "`rest/api/v1/upload_gis`", descritto nel paragrafo successivo.

5.2. RESTful Endpoint

In questo paragrafo è illustrata la lista degli endpoint REST implementati. In questi esempi tutti gli endpoint sono indicati usando come hostname il nome 'localhost', con il semplice scopo esemplificativo; sostituire 'localhost' con l'appropriato hostname o indirizzo IP del server.

5.2.1. Collaborative Platform (Piattaforma Collaborativa)

La piattaforma collaborativa è un particolare servizio che consente agli utenti di inviare notifiche geo-referenziate al web-service I-ACCESS, includendo messaggi ed immagini. La pagina di accesso alla piattaforma collaborativa è disponibile a questo indirizzo:

https://localhost:8443/coll-plat/

L'aspetto del form implementato è come quello mostrato in Figure 3

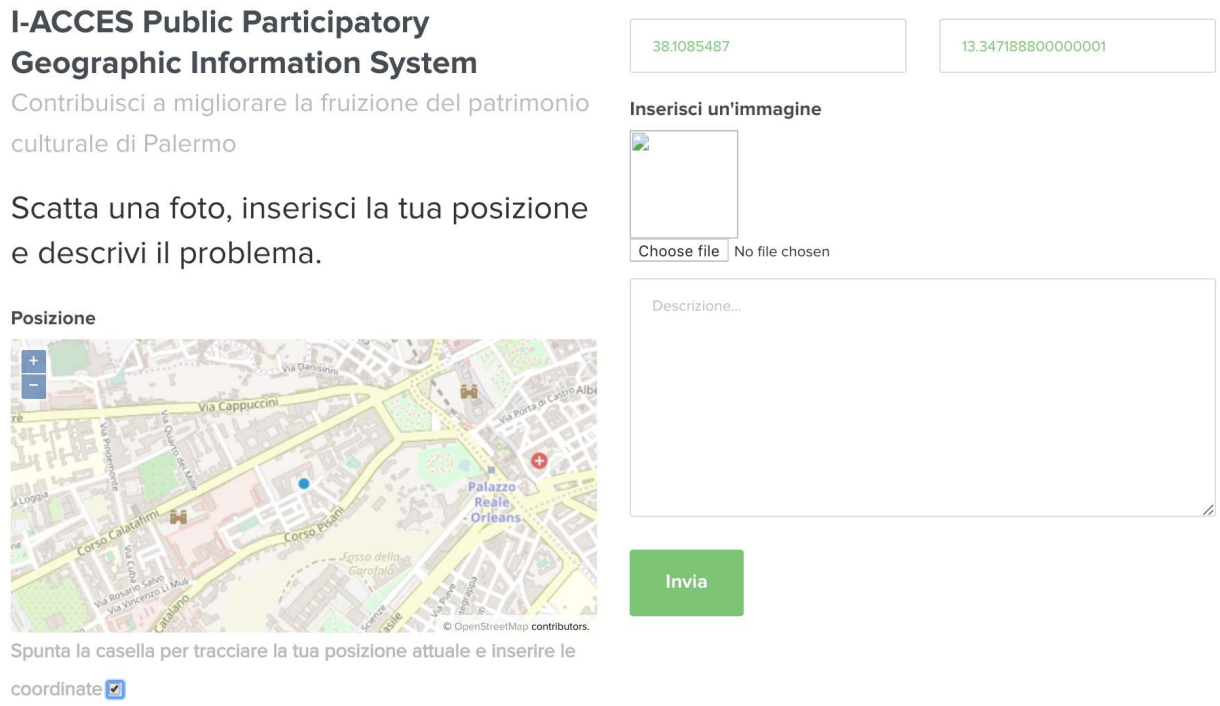

Figure 3 - Piattaforma collaborativa

5.2.1.1. Lista di tutte le notifiche sulla piattaforma collaborativa

https://localhost:8443/rest/api/v1/get_notifications

Il formato del messaggio output è in formato JSON e simile al seguente:

```
[
    {
        "geometry": {
            "type": "Point",
            "coordinates": [
                13.3472256,
                38.1083648
            ]
```

```
        },
        "id": "c53baac5-b2ab-432e-884d-bc7bce27686c",
        "description": "sdfs",
        "geom": "0101000020E6100000FFAAC88DC7B12A40CF04D4E5DE0D4340",
        "filetarget": "/img/uploads/2019-08-26 09:03:01-Schermata 2019-08-26 alle
10.40.44.png",
        "created_at": "2019-08-26 09:03:01.968242"
    },
    {
        "geometry": {
            "type": "Point",
            "coordinates": [
                13.3472256,
                38.1083648
            ]
        },
        "id": "59ad669d-77b4-486c-8284-b3aa1b6a777c",
        "description": "sdfs",
        "geom": "0101000020E6100000FFAAC88DC7B12A40CF04D4E5DE0D4340",
        "filetarget": "/img/uploads/2019-08-26 09:03:41-Schermata 2019-08-26 alle
10.40.44.png",
        "created_at": "2019-08-26 09:03:41.723739"
    },
]
```


5.2.1.2. Visualizzazione in Mappa delle Notifiche

https://localhost:8443/rest/map_notifications
L'output della mappa ha l'aspetto mostrato in figura Figura 4

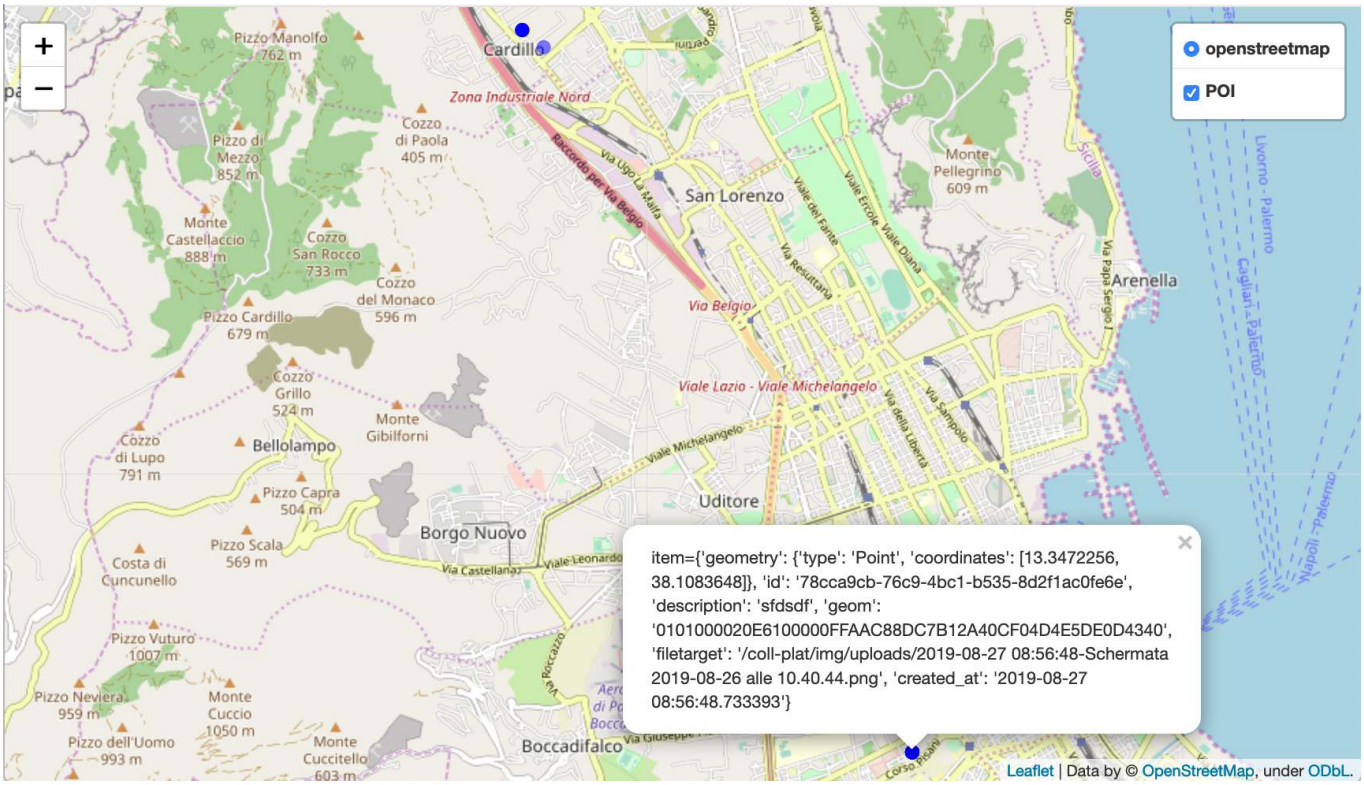

Figura 4 - Notification Map

La mappa è stata realizzata sfruttando le API di lettura delle notifications.
`rest/api/v1/get_notifications.`

5.2.2. Point of Interest (POI) and Routes

I-ACCESS offre servizi per l'accesso ai POI e ROUTES da richiamare ad applicazioni definite da terze parti.
POI list

5.2.2.1. POI list

https://localhost:8443/rest/api/v1/poi_info

L'output della chiamata REST è una lista di JSON formattata nel seguente modo:


```
[
    {
        "geometry": {
            "type": "Point",
            "coordinates": [
                14.5064745610199,
                35.8951818507155
            ]
        },
        "id_poi": "{6dd56e79-294c-4cb6-bd4a-a6ed215944c1}",
        "wchair": null,
        "vis_impair": null,
        "hear_impair": null,
        "address": null,
        "phone": null,
        "email": null,
        "url": null,
        "opening_hours": null,
        "fees": null,
        "park": null,
        "fb_url": null,
        "tw_url": null,
        "inst_url": null,
        "orig_name": "nothing before now",
        "curr_name": "test point",
        "year_costr": "2019-06-24",
        "floors": 1,
        "height_max": 10.0,
        "orig_use": "test name",
        "curr_use": "current test name",
        "owner": "mr big",
        "category": "a general place",
        "code": "0001-what",
        "geom": "0101000020E610000078967D0F93CC1B41EEC23F94A94E4E41",
        "created_at": "2019-08-26 08:38:51.589848",
        "updated_at": "2019-08-26 08:38:51.597415",
        "media": []
    },
    {
        "geometry": {
            "type": "Point",
            "coordinates": [
                14.5058599176187,
                35.8972401889024
            ]
        },
```

```
        "id_poi": "{b4d8ef3a-14db-44c8-9043-40d3094152ea}",
        "wchair": null,
        "vis_impair": null,
        "hear_impair": null,
        "address": null,
        "phone": null,
        "email": null,
        "url": null,
        "opening_hours": null,
        "fees": null,
        "park": null,
        "fb_url": null,
        "tw_url": null,
        "inst_url": null,
        "orig_name": "another place",
        "curr_name": "another place",
        "year_costr": "2019-06-24",
        "floors": 1,
        "height_max": 100.0,
        "orig_use": "place original use description here",
        "curr_use": "place original use description here",
        "owner": "mr little",
        "category": "piece of street",
        "code": "oooo1",
        "geom": "0101000020E6100000E28526CCB9CB1B41FFE456DE1B4F4E41",
        "created_at": "2019-08-26 08:38:51.589848",
        "updated_at": "2019-08-26 08:38:51.597415",
        "media": []
    }
]
```


5.2.2.2. Routes list

https://localhost:8443/rest/api/v1/route_list

L'output della chiamata REST è una lista di JSON formattata nel seguente modo:


```
[
    {
        "geometry": {
            "type": "LineString",
            "coordinates": [
                [
                    14.5076562720261,
                    35.8968362084804
                ],
                [
                    14.5087254369631,
                    35.8961624780441
                ],
```

```
                [
                    14.5090323539367,
                    35.8958492286377
                ],
                [
                    14.5095333933383,
                    35.8947352492013
                ],
                [
                    14.5107392123926,
                    35.894904718442
                ]
            ]
        },
        "id_route": "{faa99e09-9395-4762-bfa7-a5d4aea77598}",
        "code": "0001",
        "name": "first route",
        "descr": "sample route in valletta",
        "length": 0,
        "geom":
"0102000020E610000050000002FECDB85EB032D405FA56487CBF241409FCC14A977042D40038DBB73B5F2
4140288781E39F042D40A4920130ABF241402937998FE1042D40EFF244AF86F24140BA052C9C7F052D404F5
FE13C8CF24140",
        "created_at": "2019-08-26 08:38:51.624934",
        "updated_at": "2019-08-26 08:38:51.627358"
    },
    {
        "geometry": {
            "type": "LineString",
            "coordinates": [
                [
                    14.5096928120384,
                    35.8974644449814
                ],
                [
```

```
                14.5098427901108,
                35.8973252177335
            ],
            [
                14.5104169371179,
                35.8972576399418
            ],
            [
                14.5111239833561,
                35.8977725794262
            ],
            [
                14.5105121959601,
                35.8982060574969
            ],
            [
                14.5093692595303,
                35.8982898215003
            ],
            [
                14.5094566356031,
                35.8976465056942
            ],
            [
                14.5091510620057,
                35.8976061843676
            ],
            [
                14.5091510620057,
                35.8976061843676
            ]
        ]
    },
    "id_route": "{8fedf2ec-46a1-48b3-a4ba-262d748b88c5}",
```

```
        "code": "0002",
        "name": "test",
        "descr": "test",
        "length": 10,
        "geom":
"0102000020E6100000090000000286CCD74F6042D4049426C1DE0F241403B413B1D0A052D404F05808DDBF2
4140E87D685E55052D406BD19D56D9F241407B6EF10AB2052D40C9F03D36EAF241402EB7C3DA61052D40580
2856AF8F241405E8A2E0CCC042D403AD62E29FBF24140854A0980D7042D409DA9A814E6F241407213B072AF
042D40E4456BC2E4F241407213B072AF042D40E4456BC2E4F24140",
        "created_at": "2019-08-26 08:38:51.624934",
        "updated_at": "2019-08-26 08:38:51.627358"
    }
]
```


5.2.2.3. Mappa dei POI / ROUTE

https://localhost:8443/rest/map

L'output della mappa ha l'seguente mostrato in Figura 5.

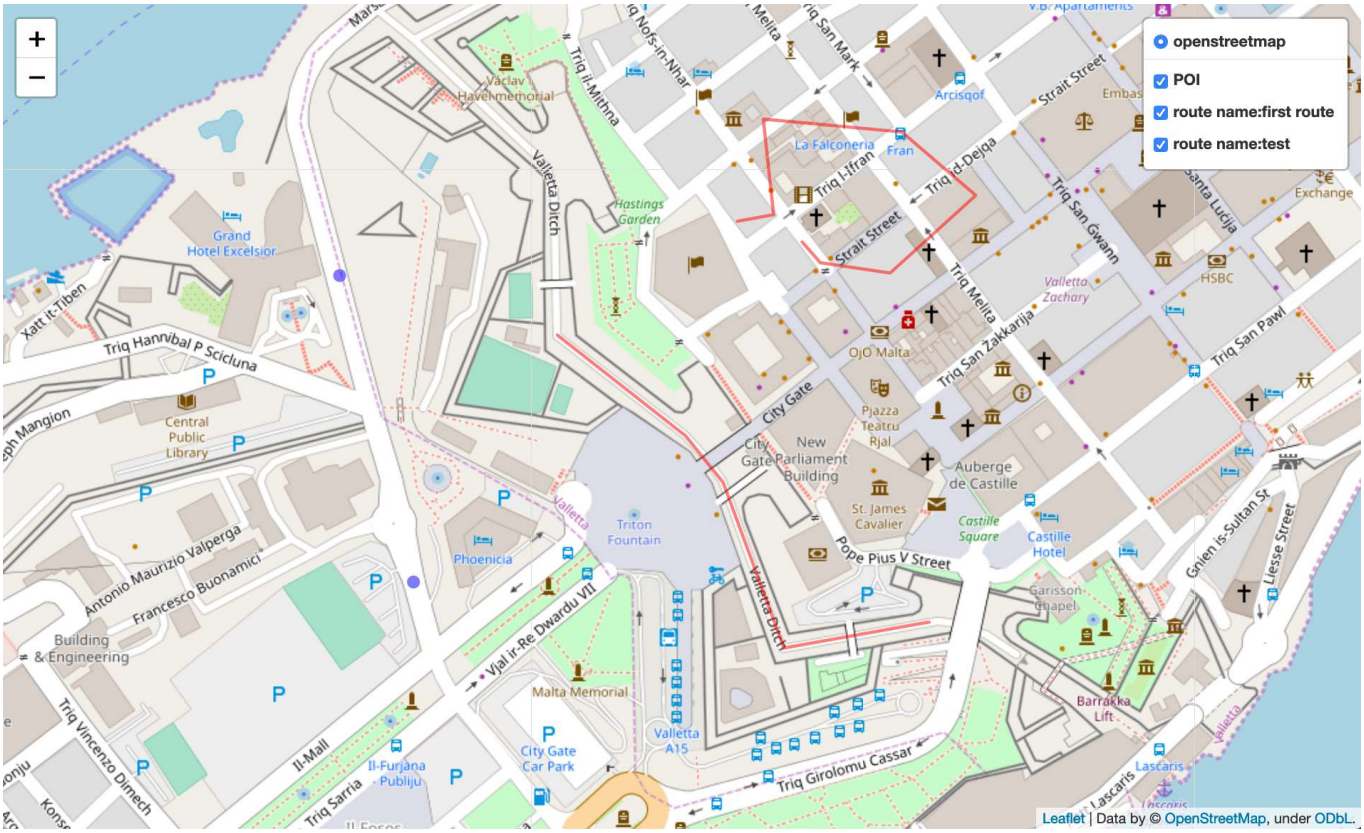

Figura 5 - POI / ROUTES MAP

5.2.2.4. Caricamento di POI, Routes ed informazioni di accessibilità

Il web-service di I-ACCESS è progettato per consentire il caricamento completo di POI, Percorsi ed informazioni di accessibilità attraverso un upload automatizzato.

Il seguente comando bash mostra un esempio di chiamata all'endpoint `rest/api/v1/upload_gis` attraverso il quale è possibile caricare un file in formato zip, opportunatamente organizzato contente le informazioni necessarie.

```
curl -X POST -H "Content-Type: multipart/form-data" -F
'gis_file=@/my/file/path/CNR_GIS.zip' https://localhost:8443/api/v1/upload_gis --user
iaccessuser:iaccess -vvv
```

Per maggiori dettagli riguardo la struttura dell'archivio suggeriamo di fare riferimento all'archivio di template ottenibile al seguente link: https://drive.google.com/drive/u/1/folders/1s9vY8RgNbBKfWb-OE3O30JiPnY7SobDs

5.2.3. Integrazione di Bluetooth Low Energy (BLE) nel web-service di I-ACCESS

Una funzionalità importante del web-service di I-ACCESS è l'integrazione di servizi di localizzazione e target advertisement prodotto mediante tecnologie Radio basate su Bluetooth Low Energy (BLE). Questa soluzione può essere usata per migliorare le soluzioni GNSS usate tradizionalmente per servizi di

localizzazione. Sono state implementate delle funzionalità di acquisizione ed analisi dei pacchetti BLE ricevuti dagli utenti.

5.2.3.1. Upload delle informazioni Bluetooth Low Energy (BLE)

Attraverso questo il codice di esempio che segue è possibile osservare l'utilizzo di un particolare endpoint chiamato "bcn_upload" il quale permette di inviare al web-service di I-ACCESS una struttura JSON che contiene le informazioni dei beacon ricevuti da. L'endpoint è stato testato sia attraverso l'applicativi di test curl (o Postman). E' stato inoltre eseguito un test di integrazione estendendo una app per android che riceve beacon di diversi formati (e.s Eddystone o I-Beacon). Il codice originario da cui è stata effettuata la modifica è disponibile su github a questo indirizzo: https://github.com/Bridouille/android-beacon-scanner.

La versione modifica è disponibile al seguente indirizzo: https://bitbucket.org/fabriziogiulianounipa/android-beacon-scanner

```
curl -k -X POST -H "Content-Type: application/json" \
-d '{
      "dev_id":"DEV_ID_VAL_EXAMPLE(eacf59b697a7245e)",
            "json_beacon":{
                  "beacon_type":"eddystone_url",
                  "distance":0.007133429116628825,
                  "eddystone_url_data":{
                        "url":"http://www.wellcoressd.com"
                  },
                  "hashcode":-1051665695,
                  "isBlocked":false,
                  "last_seen":1570436217938,
                  "manufacturer":65194,
                  "rssi":-61,
                  "tx_power":-100
            }
      }' \
      https://localhost:8443/rest/api/v1/bcn_upload \
      --user iaccessuser:iaccess -vvv
```

5.2.3.2. Acquisizione dei log salvati sul Web-service.

Attraverso questo endpoint è possibile estrarre in formato JSON una lista dei beacon ricevuti di diverso tipo (e.s. ibeacon or eddystone) ricevuti da uno specifico utente. Tra le informazioni di interesse è possibile estrarre informazioni come l'RSSI o informazioni sui messaggi (e.s. URL o informazioni dei sensori).

Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio Nazionale delle Ricerche (ICAR-CNR) Via P. Bucci cubo 8/9C, 87036 Rende (CS) – Via P. Castellino 111, 80131 Napoli – Via U. La Malfa 153, 90146 Palermo Partita IVA 02118311006 – Codice Fiscale 80054330586 – protocollo.icar@pec.cnr.it – www.icar.cnr.it

https://localhost:8443/rest/api/v1/bcn_logger


```
[
   {
      "uuid":"8e3facf8-bd76-4878-931b-cfa9e9ea6f64",
      "dev_id":"eacf59b697a7245e",
      "json_beacon":{
         "beacon_type":"eddystone_url",
         "distance":0.06428888932339942,
         "eddystone_url_data":{
            "url":"http://www.wellcoressd.com"
         },
         "hashcode":-1051665695,
         "isBlocked":false,
         "last_seen":1570531053212,
         "manufacturer":65194,
         "rssi":-76,
         "tx_power":-100
      },
      "created_at":"2019-10-08 10:37:33.408799"
   },
   {
      "uuid":"d858690e-684a-415d-8b19-5e8bf3b5345f",
      "dev_id":"eacf59b697a7245e",
      "json_beacon":{
         "beacon_type":"eddystone_url",
         "distance":0.06428888932339942,
         "eddystone_url_data":{
            "url":"http://www.wellcoressd.com"
         },
         "hashcode":-1051665695,
         "isBlocked":false,
         "last_seen":1570531053219,
         "manufacturer":65194,
         "rssi":-76,
         "tx_power":-100
      },
      "created_at":"2019-10-08 10:37:33.413293"
   }
]
```

Bibliografia

[1] Docker Container Platform, URL: https://www.docker.com/

[2] MySQL, URL: https://www.mysql.com/it/

[3] PostgreSQL Database Management System, URL: https://www.postgresql.org/

[4] PostGIS: PostgresSQL extension, URL: https://postgis.net/

[5] MongoDB, https://www.mongodb.com/

[6] PHP, URL: https://php.net/

[7] APACHE HTTP server, URL: https://httpd.apache.org/

[8] NGINX, URL: https://www.nginx.com/

[9] Python Flask mircoframework, URL: http://flask.pocoo.org/

[10] Django Project, URL: https://www.djangoproject.com/